Dynamic texture recognition and localization in machine vision for outdoor environments

Computers in Industry 98 (2018) 1–13

Contents lists available at ScienceDirect

Computers in Industry

journal homepage: www.elsevier.com/locate/compind

Dynamic texture recognition and localization in machine vision for outdoor environments

Vagia Kaltsa[a,b,*], Konstantinos Avgerinakis[a], Alexia Briassouli[a], Ioannis Kompatsiaris[a], Michael G. Strintzis[b]

[a] Centre for Research and Technology Hellas (CERTH), 6th km Charilaou-Thermi Rd, P.O. Box 60361, GR 57001 Thermi, Thessaloniki, Greece
[b] Aristotle University of Thessaloniki (AUTH), University Campus, 54124 Thessaloniki, Greece

ARTICLE INFO

Article history:
Received 15 September 2017
Received in revised form 20 December 2017
Accepted 22 February 2018
Available online 5 March 2018

Keywords:
Dynamic textures
Texture detection
Texture localization
LBP-flow

ABSTRACT

This work focuses on detecting and localizing a wide range of dynamic textures in video sequences captured by surveillance cameras. Their reliable and robust analysis constitutes a challenging task for traditional computer vision methods, due to barriers like occlusions, the highly non-rigid nature of the moving entities and the complex stochastic nature of their motions. In order to address these issues, a novel hybrid framework is introduced, combining representations on both a local and global scale. A new, handcrafted local binary pattern (LBP)-flow descriptor with Fisher encoding is initially used to effectively capture low level texture dynamics, and a neural network (NN) is deployed after it to obtain a higher level, deeper and more effective representation scheme, capable of robustly discriminating even challenging dynamic texture classes. A novel localization scheme, based on multi-scale superpixel clustering is introduced, in order to detect texture patterns on local and global scales, inside and throughout sequential video frames. Experiments on various challenging benchmark datasets prove our method's efficacy and generality, as remarkable recognition and localization accuracy rates are achieved at a low computational cost, making it appropriate for real world outdoor applications.

1. Introduction

Dynamic texture recognition, localization, and more generally dynamic scene analysis in videos constitutes an intriguing topic within the computer vision community, due to its wide applicability in many scenarios. The term dynamic texture typically refers to moving textures, i.e. visual entities undergoing small, stochastic motions, encountered in real world indoor and outdoor environments. Current work mainly focuses on outdoor scenarios where a crisis event might occur (i.e. fire in a forest, a flooded river etc.), so we mostly examine classes of this category, even though, several instances of dynamic textures appearing in indoors videos, are also examined so as to prove our algorithm's efficacy and generalization The automatic recognition of such textures has recently attracted attention, as it can provide a significant contribution to many real-world outdoor applications involving: scene analysis containing objects with high varying textures (e.g. water, smoke, trees), security applications for the prevention of a possible terrorist act and surveillance systems, responsible for the avoidance of natural disasters (e.g. fire in the forest or floods).

The main challenges for the analysis of dynamic textures and scenes, especially those taking place outdoors are: (a) illumination changes, (b) complex and unpredictable motion patterns, (c) occlusions, (d) the presence of rigid and non-rigid objects in the same scene, (e) camera motion, and finally (f) significant intra-class differences among patterns of the same category. Computational efficiency is an additional factor, as it has to be kept within reasonable limits, so as to be used by real-world applications.

Even though several methods have been proposed in order to discriminate and classify dynamic textures, they are usually restricted to a global video classification approach, neglecting the necessity of a local aspect, which may prove to be of vital importance in an emergency situation (e.g. flood). In order to address this issue, we introduce a novel hybrid framework involving both dynamic texture recognition and localization in outdoors videos captured by surveillance cameras. The LBP-flow descriptor, introduced in our previous work [1], is further investigated and applied to effectively capture textures' motion dynamics while a principal component analysis (PCA)-Fisher

* Corresponding author at: Centre for Research and Technology Hellas (CE.R.T.H.), Information Technologies Institute, Building A – Office 2.4, 6th km Xarilaou – Thermi, 57001 Thessaloniki, Greece.
E-mail address: vagiakal@iti.gr (V. Kaltsa).

https://doi.org/10.1016/j.compind.2018.02.007
0166-3615/

scheme is used to address the issues of efficient encoding and handling of high dimensional data. A neural network (NN) is fed with the outcome of Fisher encoding, providing a deeper representation of the dynamic texture. Both binary and multi-class classification models are acquired and are used in our novel localization scheme, based on multi-scale superpixel clustering to spatio-temporally detect dynamic textures in videos. The overall recognition and localization framework is depicted in Fig. 1.

Our contribution to recognition and localization of texture dynamics can be summarized as follows:

1 A multi-scale superpixel-based clustering scheme is introduced, to achieve balance between local and global features. Local characteristics are retained via the superpixels, and clustering allows the capture of more global patterns. In this manner, our method avoids overfitting to local noise and succeeds in obtaining hybrid global-local descriptors for accurate texture localization in unsegmented video samples.
2 Novel pre-trained hand crafted descriptors are developed, leading to a near real-time recognition and localization framework, appropriate for real world applications.
3 Neural networks are applied on our hand crafted descriptors, resulting in improved higher level descriptors and increased recognition accuracy.

2. Related work

Dynamic texture recognition methods can roughly be separated into two main categories according to their adopted underlying model. The first category refers to *Generative models* which involve the extraction of global features throughout video sequences and their modeling is based on some hidden parameters [2]. Recent works such as Doretto et al. [3] use the spatio-temporal dynamics to train a Gauss-Markov recognition model, while Chan et al. [4] propose an expectation maximization (EM) algorithm to train the parameters of a statistical model. In [5] a linear dynamic texture (LDT) scheme is proposed in order to represent a stochastic model of different appearance and motion dynamics. Lately, linear dynamical systems (LDS) raised a lot of attention within this category, with the work of [6] being a representative example. In their work, an hierarchical EM algorithm is deployed in order to cluster and learn the statistical model of the motion dynamics. LDS has recently been extended into a stabilized higher order LDS (shLDS) in [7], who introduced Histograms of Grassmannian Points (HoGP). However, despite its high accuracy rates the method is computational costly, making it inappropriate for real-time applications.

While generative models seem quite promising for representing dynamic textures, their application to classifying the wider set of motion patterns found in dynamic scenes has been shown to perform poorly [8]. The complex, stochastic character of dynamic textures makes their precise modeling very challenging, so a second category of dynamic texture representation, namely *Discriminative models* has been considered. This category is based on the extraction of local, spatio-temporal features to describe moving texture dynamics by estimating local variations and statistics of intensity and optical flow values. Early techniques involved the accumulation of local spatio-temporal features using appearance features like GIST [9], motion histograms, such as the Histograms of Oriented Optical Flow (HOOF) [10], swarm-intelligence [11], spatio-temporal oriented energy features (STOEF) [12], and their successful and highly accurate Bag-of-Words(BoW) extension proposed in [13], named spatial energies. However, the coarse quantization of GIST and the rotation invariance of HOOF do not allow them to detect dynamic textures with accuracy, while on the other hand, the highly accurate STOEF, spatial energies and swarm dynamics suffer from computational efficiency making them inappropriate for real case implementations, such as surveillance and security scenarios.

Accurate texture classification has been achieved in images using local binary patterns (LBPs), whose promising results have led to a number of its extensions as a dynamic texture descriptor. Volume local binary patterns (VLBP) [14] and LBP-TOP [15] are among the earlier methods, however they can easily reach a dimensionality of 2^{14}–2^{26}, which is impractical in real-world applications involving large amounts of data that are to be processed in near real time. Mettes et al. [16] has recently introduced a hybrid spatio-temporal extension of LBP, which stacks the descriptor in time to obtain temporal information. Even

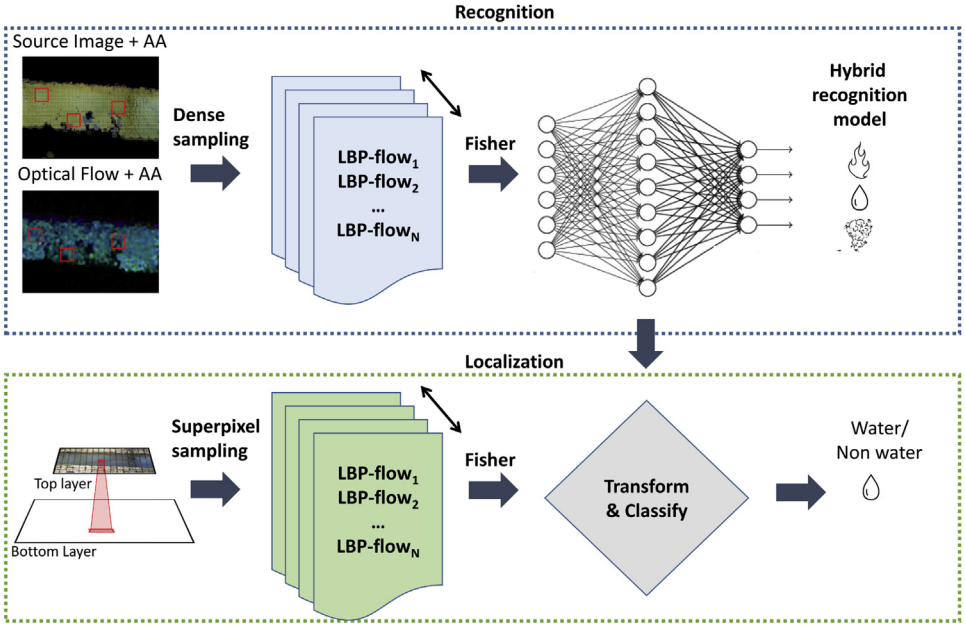

Fig. 1. The overall recognition and localization framework.

though, the method achieved very high accuracy rates when discriminating between water and non-water scenes, its highly tailored character to exclusively water class, makes it inappropriate for more general classification and localization scenarios.

In this work, we overcome the limitations of the SoA, by using the proposed LBP-flow descriptor. Then, a PCA-Fisher scheme is applied, reducing final descriptor's dimensionality, while increasing its discriminative ability, and the outcome is fed into two different learning models, with both shallow and deep schemes used for validation, for binary and multi-class recognition purposes. A superpixel clustering structure is then used for dynamic texture localization and it is deployed in challenging real world videos, including videos of high interest for outdoors surveillance, to evaluate the efficacy and applicability of our framework. The robustness and efficacy of the proposed framework is proven in experiments on benchmark datasets, where high accuracy is achieved in all tasks and at a low computational cost.

There exist several works in the literature that address the problem of spatio-temporal segmentation of dynamic textures. More specifically, in the work of Amiaz et al. [17] videos are split into regions obeying different models based on brightness constancy and brightness conservation, while Doretto et al. [3] deploy Gauss-Markov models in order to represent region dynamics and to group regions with similar spatiotemporal signatures. A Markov Random Field based approach is adopted in [18] to partition a video into disjointed regions with dynamic textures, showing similar uniformity and consistency, while appearance and motion information is used by [10] in a distance-related framework to efficiently segment video sequences. Finally, a statistical model is proposed by Chan et al. [4] in which each video sequence is modeled as a collection from a set of underlying dynamic textures, resulting in efficient video clustering and segmentation.

However, it should be noted that in all these works, dynamic texture segmentation essentially refers to the extraction of regions with a similar kind of motion, and not necessarily the same kind of dynamic texture. For this reason, in our work, we prefer to use the term “localization” instead of “segmentation”, in order to emphasize our differentiation from the literature, where the term “segmentation” is often used to describe motion-based methods. Dynamic texture segmentation usually refers to the extraction of regions with homogeneous dynamics, regardless of the texture involved. However, the same dynamic texture class may feature different motion characteristics in different parts of it. For example, a scene with a fountain comprising of different water dynamics would be split into several areas with homogeneous motion characteristics. Our framework has a different goal: its aim is to identify each dynamic texture class (or category, e.g. fire, water etc) as a single cohesive area. Thus, instead of segmenting or clustering the scenes into areas with similar spatio-temporal properties, we focus on dynamic texture classification combined with texture localization. In the literature on dynamic texture segmentation, we have found a similar approach to ours only in [16], however that approach is highly tailored to the water class for both classification and localization tasks, and cannot be applied to other types of dynamic textures.

The paper is organized as follows: Section 3 describes spatio-temporal representation of dynamic scenes; Section 4 presents the texture detection and localization framework; while a detailed experimental evaluation is discussed in Section 5. Finally, conclusions are summarized in Section 6.

3. Spatio-temporal representation of dynamic textures

In order to effectively deal with the challenging nature of videos containing outdoors unconstrained environments, their representation should be firstly carefully examined and determined. The stochastic movements of the ensembles comprising dynamic textures in combination with their non-rigid nature, require the adoption of general descriptors, capable of managing highly unpredictable and ambiguous types of videos. To this end, we adopt the new LBP-flow descriptor, which is then encoded by Fisher vectors resulting in an informative mid-level descriptor. The process is shown to be able to accurately classify dynamic scenes whose complex motion patterns are difficult to separate otherwise. LBP-flow and Fisher encoding used for dynamic texture representation are discussed in more detail in the sections that follow.

3.1. LBP-flow

The new descriptor LBP-flow introduced in our previous work [1] was adapted and further investigated in order to accurately describe videos' underlying structure, as it has proven to effectively encode both appearance and motion induced variations, present in dynamic textures. LBP-flow constitutes an extension of the well known LBP [19], which was chosen due to its successful applicability in a variety of texture classification ([20,21,15,14]) and face recognition tasks ([22,23]). Thus, inspired by its success, LBP-flow builds upon the original LBP and extends it over time providing a powerful shallow spatio-temporal descriptor.

In classic LBP, the LBP value of a particular pixel $\bar{r}$ is computed by comparing its intensity value with that of its neighboring pixels. LBP-flow extends this definition to also include the values of the optical flow around pixel $\bar{r}$, so as to embed motion information. More precisely, the new LBP-flow values of a pixel $\bar{r}$ are given by:

$$LBP-flow(\bar{r}) = \sum_{p=0}^{P-1} s(f(\bar{r}) - f(\bar{r}_p))2^p \quad (1)$$

where $f(\bar{r})$ corresponds to either pixel's intensity or optical flow value, P represents the number of neighbor pixels around each sampled pixel $\bar{r}$, and $(\bar{r}_p)$ stands for the neighbor points around pixel $\bar{r}$ in radius R, at coordinates $\bar{r}_p = (r_x + R \cos(2\pi p/P), r_y - R \sin(2\pi p/P))$. The threshold function $s(z)$ of LBP-flow is given by:

$$s(z) = \begin{cases} 0, & z < 0 \\ 1, & z \geq 0 \end{cases} \quad (2)$$

The novelty of LBP-flow is that it represents both intensity and optical flow variations over space and time. In our LBP representation, texture is spatially represented by calculating local binary patterns in two directions, on the $x-y$ axes, as in the original LBP. A novel representation of motion as a temporal texture is introduced by calculating LBP over the optical flow values in the x and y directions, $x-t$ and $y-t$ respectively. This inclusion of motion information in the LBP-flow representation enriches the descriptor's spatio-temporal characteristics leading to a more robust and efficient shallow representation. By this procedure, we obtain the LBP-flow descriptors for appearance and motion, namely LBP_{xy}, LBP_{xt} and LBP_{yt} respectively.

The dimensionality of the resulting LBP_{xy}, LBP_{xt} and LBP_{yt} is then reduced by using a variation of the original LBP descriptor, the uniform quantized LBP descriptor [15], which uses 58 bins to describe a 3×3 area around each interest point instead of the 256 bins commonly used. To achieve this, uniform quantized LBP takes into account that there is one quantized pattern for each LBP, with exactly one transition from 0 to 1, and one from 1 to 0 when scanned counter-clockwise. Thus, the uniform quantized LBP represents the same pattern with a descriptor whose dimension is equal to 1/4 of the original LBP.

The LBP-flow descriptor is finally constructed by accumulating LBP_{xy}, LBP_{xt} and LBP_{yt} over a time window of W frames, followed by

their concatenation into a single vector. Thus, for a pixel $\bar{r}$, the final LBP-flow descriptor is given by:

$$\{LBP^{\bar{r}}_{xy1},\ldots,LBP^{\bar{r}}_{xyW},LBP^{\bar{r}}_{xt1},\ldots,LBP^{\bar{r}}_{xtW},LBP^{\bar{r}}_{yt1},\ldots,LBP^{\bar{r}}_{ytW}\} \quad (3)$$

The representation framework of the LBP-Flow is depicted in Fig. 2.

3.2. Fisher encoding

LBP-flow includes rich spatiotemporal information as a low-level local representation, but also allows for redundancies, such as intra-class pattern deviations and noise-induced artifacts. In order to constrain this noise and subsequently increase the discriminative ability of our descriptor, the Fisher Vector representation is adopted, transforming initial LBP-flow vectors of each video sample into a mid-level single vector representation, based on the detected most discriminating features (visual vocabulary) of the overall video-samples. In this way, the size of our descriptor is significantly reduced, while at the same time recognition accuracy is increased. The computation of the most discriminating samples is performed by applying unsupervised clustering (Gaussian Mixture Model (GMM)) in the shallow representation hyperspace, as formed by the LBP-flow feature collection of each dynamic texture dataset. Let $\{\mu_j, \Sigma_j, \pi_j; j \in R^L\}$ be the set of parameters for L Gaussian models, with μ_j, Σ_j and π_j standing respectively for the mean, the covariance and the prior probability weights of the jth Gaussian. Assuming that the D-dimensional LBP-flow descriptor is represented as $\bar{x}_i \in R^D; i = \{1,\ldots,N\}$, with N denoting the total number of descriptors, Fisher encoding is then built upon the first and second order statistics:

$$\begin{aligned} f_{1j} &= \frac{1}{N\sqrt{\pi_j}}\sum_{i=1}^{N} q_{ij}\sigma_j^{-1}(\bar{x}_i - \bar{\mu}_j) \\ f_{2j} &= \frac{1}{N\sqrt{2\pi_j}}\sum_{i=1}^{N} q_{ij}\left[\frac{(\bar{x}_i - \bar{\mu}_j)^2}{\sigma_j^2} - 1\right] \end{aligned} \quad (4)$$

where q_{ij} is the Gaussian soft assignment of descriptor x_i to the jth Gaussian and is given by:

$$q_{ij} = \exp\frac{\left[-\frac{1}{2}(x_i-\mu_j)^T\Sigma_j^{-1}(x_i-\mu_j)\right]}{\sum_{t=1}^{L}\exp\left[-\frac{1}{2}(x_i-\mu_t)^T\Sigma_j^{-1}(x_i-\mu_t)\right]} \quad (5)$$

Distances as calculated by Eq. (4) are next concatenated to form the final Fisher vector, $F_X = [f_{11}, f_{21}, \ldots, f_{1L}, f_{2L}]$, characterizing the dynamic texture of each video. A Hellinger kernel $K(g, h)$ is then chosen to be used for the normalization of the vectors, due to its increased sensitivity to smaller bin values, leading to SoA results [24]. Given two Fisher vectors $g \in R^{2LD}$ and $h \in R^{2LD}$, $K(g, h)$ is computed by:

$$K(g,h) = \sum_{j=1}^{2LD} sign(g_j)sign(h_j)\sqrt{\|g_j\|\cdot\|h_j\|} \quad (6)$$

where L is the number of Gaussians, D the dimensionality of the descriptor and $2LD$ the final Fisher vector size.

4. Dynamic texture recognition and localization

Given the aforementioned powerful descriptor, a framework is proposed for dynamic texture recognition and localization. Fisher vectors either are used to train a binary/multi-class support vector machine (SVM) classifier or a neural network (NN), in order to learn to discriminate between two or more classes. The framework including the NN can be characterized a hybrid representation scheme, as it leverages both shallow and deep parameters to train a final classification model. Dynamic texture localization follows, to spatio-temporally localize the selected dynamic texture inside, and throughout, sequential video samples. The proposed scheme exploits the resulting binary model of the aforementioned recognition process and based on a superpixel clustering procedure leads to an accurate and computationally efficient localization framework. Dynamic texture recognition and localization processes are discussed in more detail in the following subsections.

4.1. Dynamic texture recognition

Dynamic texture recognition requires an accurate sampling process, so as to collect a sufficient number of informative feature samples to train the discriminative model. In our framework, activity areas (AA) [25] are used as an initial step to detect regions of interest and to sample interest points in them to be used for training purposes. AA are binary masks, extracted according to the premise that flow estimates originate either from actual motion, or noise, e.g. from the video capture or compression process. These two hypotheses can be formulated as:

$$\begin{aligned} H_0 &: u_k^0 = z_k(\bar{r}) \\ H_1 &: u_k^1 = u_k(\bar{r}) + z_k(\bar{r}) \end{aligned} \quad (7)$$

where $\bar{r} = (x, y)$ is the pixel under consideration, $u_k(\bar{r})$ its actual motion value and $z_k(\bar{r})$ is induced by noise. As shown in [25], $z_k(\bar{r})$ can be modeled by a Gaussian probability density function when optical flow is taken into account for the computation of motion vectors. Under the Gaussian assumption for the noisy flow values, and accumulating these motion vectors over a temporal window W_{AA}, we can easily eliminate them by estimating their Kurtosis, which is equal to zero for Gaussian data. Thus, the AA has zero values at pixels where the kurtosis values tend to zero, corresponding to noise induced cases, and finally resulting in a

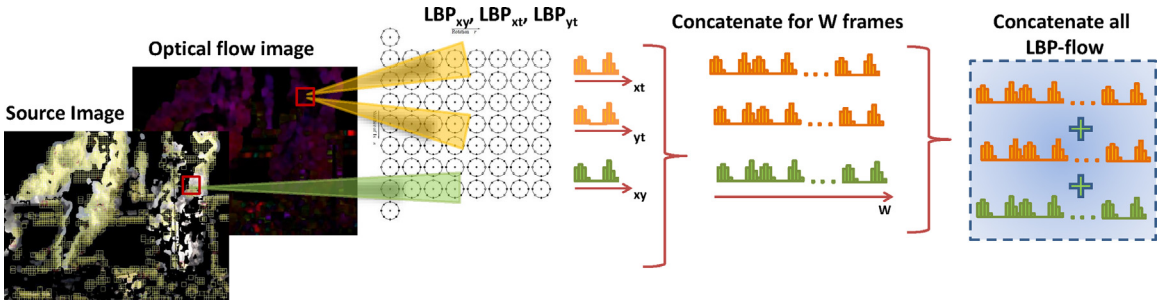

Fig. 2. LBP-flow computation.

binary mask that effectively separates background from foreground areas.

$$AA(\bar{r}) = \begin{cases} 0 & \text{if } G_2(\bar{r}) < th_{AA} \\ 1 & \text{else} \end{cases} \quad (8)$$

where th_{AA} is statistically determined equal to $2 \cdot 10e^{-2}$ based on the experiments performed on Dyntex dynamic texture dataset.

LBP-flow as described in Section 3.1 is then calculated over a block of 32×32 pixels, around each interest point. Subsequently, Fisher encoding is deployed after a PCA dimensionality reduction step and the total set of descriptors representing the whole training corpus are led as an input into two different learning models, for comparison reasons. The first one, concerns the classic support vector machine (SVM) model which has proven to exhibit good performance relatively to other machine learning methods, while at the same time being fast to run and capable of handling large data sequences, generally appearing in real life situations. The second model, refers to the adoption of a neural network (NN) scheme as inspired from the successful results presented in [26] consisting of three hidden layers, each of which is followed by a dimensionality reduction step. The statistical power that Fisher vectors encapsulate in their scheme, passes in the NN as well and leads to a highly discriminative vector as shown in Section 5.

The choice of NNs over deep learning features is justified by the fact that even though deep learning features have proven to be highly accurate for image recognition tasks, the current SoA is turning its attention to more flexible hybrid patterns that leverage low level representation schemes so as to improve their results and provide a faster solution ([24,26]). As opposed to multi-layer deep learning schemes, Neural Networks are smaller and simpler, without losing their discrimination power. We thus deploy a novel representation scheme that is based on shallow spatio-temporal descriptors, encoded with a Fisher vector scheme and enhanced with a neural network learning framework so as to classify dynamic texture patterns.

The block diagram of the texture recognition framework is depicted in Fig. 3.

4.2. Dynamic texture localization

Although the aforementioned descriptor can effectively capture scene dynamics for video classification, the adoption of a local approach is needed in order to achieve accurate localization of a dynamic texture within a video frame. For this purpose, a multi-scale superpixel scheme is proposed, as superpixels enable the grouping of pixels into regions with a homogeneous appearance, which are highly likely to correspond to the same object. Furthermore, this process also eliminates redundant image information, leading to the extraction of more accurate object contours.

In our implementation, superpixels, extracted according to the simple linear iterative clustering (SLIC) method in [27], are used to segment the video frames. SLIC is based on a local version of *K*-means algorithm, where the only parameter that needs to be specified is k, which stands for the number of approximately equally sized superpixels. Then, an iterative 2-step process begins with each pixel being assigned to its nearest cluster center followed by the computation of the residual error between the new cluster center and previous cluster center locations as derived from L2-norm. This process is repeated until convergence. The distance measure used for the clustering is based on pixels' color and location and is given by:

$$D = \sqrt{d_c^2 + \left(\frac{d_s}{S}\right)^2 m^2} \quad (9)$$

where d_c and d_s stand for color similarity and spatial proximity respectively, S is the grid sampling interval and m constitutes a relative importance weight between color and spatial proximity. Thus, larger m favors spatial proximity leading to more compact superpixels, while smaller values of the variable result into superpixels with less regular size and shape, but boundaries that are more close to the image itself. Finally, a post-processing step to cluster some remaining individual "orphaned" pixels takes place by using a connected components algorithm.

Superpixels are then deployed in a 2-layers scheme, with each layer corresponding to a different scale, following a fine to coarse structure. This way, both coarse and fine details are successfully captured, and the influence of local noise is avoided. Next, we carry out what we refer to in this work as *superpixel clustering*, which relates each superpixel in the top coarser layer with multiple superpixels of the finer bottom layer according to the overlap they have with each other, and a final descriptor characterizing the whole area covered by the superpixel of the top layer is extracted.

The superpixel clustering procedure is as follows: let s_1 be the superpixel of the top layer and $s_2, s_3, \ldots, s_n$ be the superpixels of the bottom layer that are related to s_1, based on their regions' overlap. LBP-flow is calculated in a block of 32×32 pixels around all n superpixels' centers, and all features are concatenated to form the descriptor characterizing the area covered by s_1. In order to account for temporal variations, which are central in the dynamic textures, the video sequence is also divided into non-overlapping subsequences of W frames, as described in Section 3.1 and thus, the descriptor for each area A_1 covered by a top layer's superpixel s_1 is given by:

$$descr_{A_1} = \begin{Bmatrix} LBP^{s1}_{xyW}, LBP^{s1}_{xtW}, LBP^{s1}_{ytW} \\ LBP^{s2}_{xyW}, LBP^{s2}_{xtW}, LBP^{s2}_{ytW} \\ \vdots \\ LBP^{sn}_{xyW}, LBP^{sn}_{xtW}, LBP^{sn}_{ytW} \end{Bmatrix} \quad (10)$$

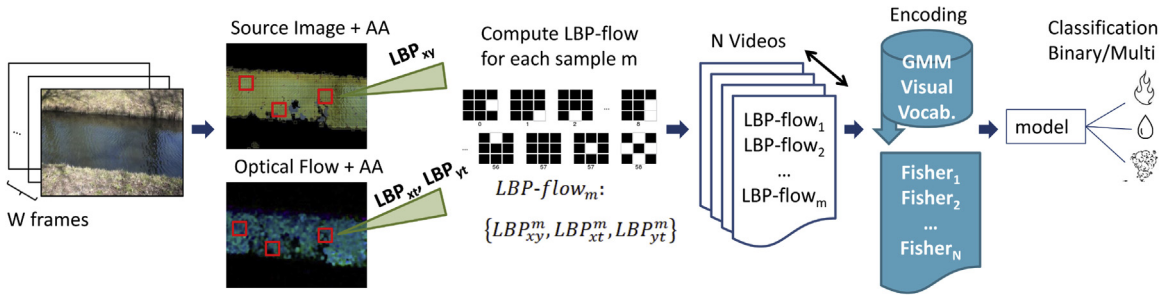

Fig. 3. Block diagram of the texture recognition framework: activity areas (AAs) are applied on each video frame and optical flow matrix, so as to sample and aggregate LBP descriptors in the three spatio-temporal domains (x, y, t). LBP-flow final descriptors are then fed to a GMM to compute the visual vocabulary and Fisher vectors are subsequently extracted for each video sample, based on this vocabulary. Results are given to a learning model responsible for their classification.

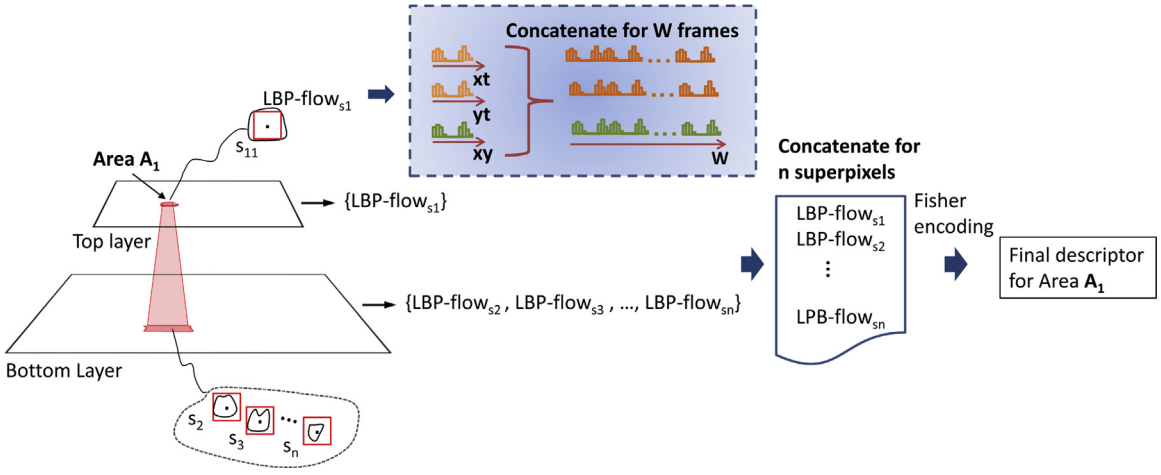

Fig. 4. Extraction of the final descriptor representing the area A_1 corresponding to superpixel s_1 of the top layer. LBP-flow is calculated in a block of 32×32 around the center of each superpixel, for a time window of W frames. Then, LBP-flow descriptors of superpixel s_1 and those of superpixels $s_2, s_3, \ldots, s_n$ belonging to the bottom layer and found in the range of s_1, are concatenated. Fisher encoding is next applied to the concatenated features and the final descriptor corresponding to the area A_1 is obtained.

where LBP^{sn}_{xyW} denotes the spatial dimension of LBP-flow for the supervoxel corresponding to superpixel *sn* for W consecutive frames, while LBP^{sn}_{xtW} and LBP^{sn}_{ytW} stand for the other two motion dimensions of LBP-flow respectively, concerning the supervoxels of the same superpixel *sn*, at a time window of W frames. PCA dimensionality reduction and Fisher encoding is then applied in the resulting LBP-flow, leading to the final descriptor characterizing the particular area in the frame. The overall procedure is shown in Fig. 4. Our intuitive expectation is that the use of multiple resolutions will result in more accurate localization while excluding multiple miss-classifications, occurring in small areas.

After the extraction of area's descriptor, the discriminative models that have been trained in the aforementioned binary classification task, are used in order to localize the desired dynamic texture in a spatio-temporal manner. The decision is conducted locally for each area covered from superpixels of the top layer. The complete localization scheme is depicted in Fig. 5.

5. Experiments

In order to evaluate the effectiveness of our method, we have applied it on four challenging benchmark datasets, namely Dyntex [28], MovingVistas [8], Yupenn [12] and videoWaterDatabase [29]. All datasets were split into 1/3 for testing and 2/3 for training, creating 3 different train/test splits to assess the performance of our algorithm. In all cases, our algorithm's accuracy was calculated in multiple tasks and compared with the SoA, demonstrating improved performance.

5.1. Parameter estimation

For the classification task, a time window of $W_{AA} = 30$ frames is used for the calculation of activity areas, where motion is estimated using Färneback optical flow [30], as it combines high accuracy and computational efficiency, making our method suitable for real world scenarios. In this implementation of optical flow, local regions of each video frame are approximated by polynomial expansion. The size of the neighborhood to be modeled plays the role of a regularizer, as modeling over larger regions results in smoother approximations, making singularities less likely. In our work we used the typical parameters *poly _ n* = 5 and *poly _ s igma* = 1.2, which were found to achieve the best trade off between motion estimation accuracy and computational speed. Our results, using these parameter values, were found to be accurate over a range of videos, so they remain the same during all experiments. Additionally, the choice of this algorithm was empirically found to result in accurate foreground segmentation in the form of Activity Areas, as well as capturing small motions, which occur often in dynamic textures.

The temporal window used is also determined experimentally so as to be maximally informative and at the same time achieve a balance between retaining useful information and being robust to noise. More specifically, the size of the temporal segments is specified at 15 and 30 frames for recognition and localization tasks respectively, so as to keep them as small as possible, which is appropriate for real-world applications, but at the same time long enough to capture texture dynamics. Experiments on even smaller

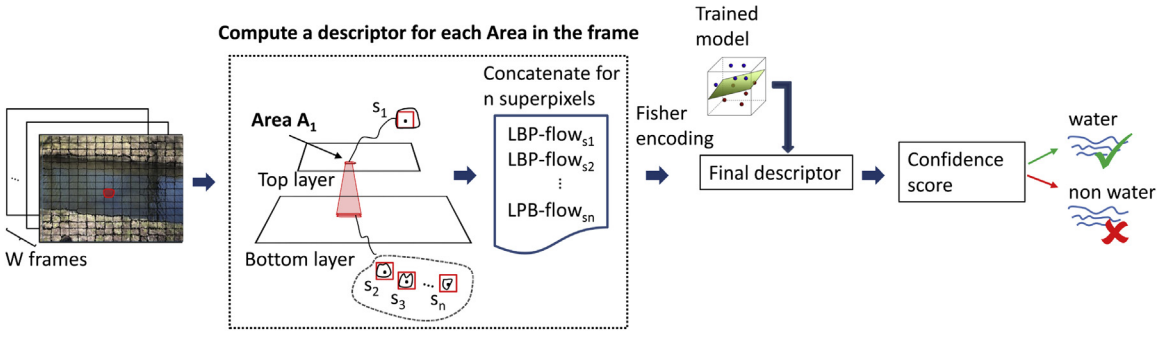

Fig. 5. Block diagram of the overall localization framework in case of the water class: Superpixel clustering is applied on each video frame and descriptors are calculated for each area. Then, the same trained model formed from the detection task is applied to each descriptor and the localization process of the desired dynamic texture takes place.

windows, such as 10 frames, were also conducted for the localization task, to determine if this would lead to a faster accurate approach. However, the results provided in Table 7 demonstrate that using a 10 frame time window leads to less accurate results, but without a significant increase in algorithm's speed. Therefore, we prefer to use a 30 frame window in practice for the localization task, as better accuracy is achieved at only a slightly higher computational cost, making it more appropriate for reliable and accurate predictions.

Thus, the extraction of the LBP-flow descriptor in a window of $W_{LBP} = 15$ or $W_{LBP} = 30$ frames, results in a feature vector of 2610 or 5220 bins respectively. These dimensions are much lower than those of the latest SoA LBP [15], which reach dimensions of 2^{14} and 2^{26}, having a positive impact in our algorithm's final speed. The dimension of the initial vector LPB-flow is reduced via PCA to 80 and 32 cluster centers are chosen to represent the visual vocabulary. Subsequently, two different learning models are tested: (a) a multi-class linear SVM which constitutes the shallow approach and (b) a neural network (NN) with 3 hidden layers, each of which is followed by a dimensionality reduction step, and a softmax output, constituting the hybrid method. For multi-class classification we found that 512 neural nodes in the first, 256 and 128 nodes in the second and third layer respectively were sufficient to create an accurate recognition model, while a dropout of 0.5 was chosen in all cases. In order to validate the performance of our proposed scheme, we provide the classification accuracy for both models, as well as comparisons with SoA.

During the localization process, the size of the superpixels' grid is determined based on the trade off between capturing sufficiently noise-free local information and keeping the computational cost of the algorithm as low as possible. The addition of a 3rd layer increased the computational cost to prohibitive levels, while the use of a single layer was found to be inadequate for accurate localization, making the option of two layers an ideal choice. The determination of the grid in each layer was based on the same concept: an 8×8 grid proved to be too coarse, as texture boundaries were not well defined, while a finer grid of 64×64 led to many noise artifacts, with the run time also significantly increasing. As a result, grids of 32×32 and 16×16 were deployed for the bottom and top layer respectively. This way, both coarse and fine details are successfully captured while at the same time local noise is avoided.

It should be emphasized that in all the cases examined in our work, irrespective of the datasets used, the above parameters remain the same, reinforcing our claim about the generality of the proposed algorithm.

5.2. Evaluation criteria

In order to evaluate our method in dynamic texture recognition, we conducted several experiments, providing confusion matrices and the average accuracy rates, which are compared with the corresponding SoA. In the localization task, we assess our method's performance by using the metric from [16], where the detection fit D of a binarized video V compared to a ground truth mask M is given by:

$$D(V, M) = \frac{\sum_{i=1}^{|V|} d(V_i, M)}{|V|} \tag{11}$$

where $|V|$ denotes the number of frames in V, V_i stands for the ith frame, and $d(V_i, M)$ is defined as:

$$d(V_i, M) = 1 - \frac{\sum_{x=1}^{W} \sum_{y=1}^{H} |V_i(x, y) - M(x, y)|}{W \cdot H} \tag{12}$$

with W and H denoting frame's width and height respectively.

5.3. Dynamic texture recognition

5.3.1. DT Recognition:The DynTex dataset

Dyntex [28] is one of the earliest and most renowned benchmark datasets for dynamic textures, containing a wide variety of texture classes. In our experiments, we use the benchmark classification split of DynTex dataset into three subsets: alpha, beta and gamma. These subsets contain video samples from 3, 10 and 10 different classes respectively, often including high intra-class variance. Some instances of DynTex dataset are shown in Fig. 6.

The overall average score of the method proposed is provided in Table 1, where it can be seen that our work outperforms the SoA in all 3 subsets. More specifically, our method is compared against 6 other SoA works including: the wavelet decomposition approaches of [31] and [32], the use of dynamic fractal spectrum (DFS) of [33], the deployment of LBP-TOP [15], the use of SIFT-like feature descriptors of [34] and finally the wavelet-based multi-fractal spectrum (WMFS) encoding of [35]. Our results for both shallow (SVM) and hybrid (NN) schemes are presented, with the latter achieving remarkably higher scores, exceeding 97% in all cases. The confusion matrices for SVM and NN based classification are provided in Fig. 7-a and-b, respectively. As shown, our descriptor achieves high accuracy for all classes, while high accuracy, over 95% is reached for 8 out of 10 classes in our NN-framework. In the lower-right part of Fig. 7, it can be seen that misclassifications of water-related classes usually refer to other classes related to water

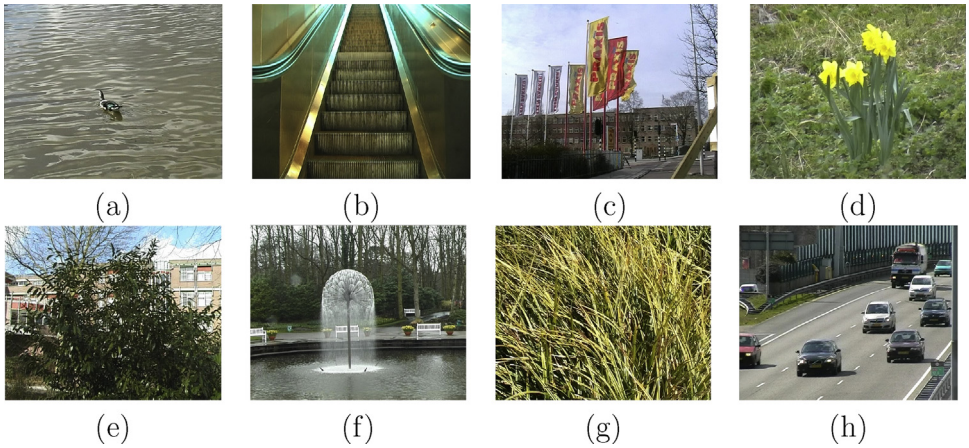

Fig. 6. Instances of DynTex dataset.

Table 1
Comparisons with SoA in DynTex dataset for alpha, beta and gamma splits.

	Dubois et al. [31]	Smith et al. [32]	DFS [33]	LBP-TOP [15]	OTF [34]	WMFS [35]	Ours SVM/NN
Alpha	88%	83%	85.2%	83.3%	83.6%	84.8%	95.0/**100%**
Beta	66%	67%	76.9%	73.4%	73.2%	75.2%	94.8/**97.4%**
Gamma	65%	65%	74.8%	72%	72.5%	73.3%	95.6/**98.0%**

Bold values indicate the highest score achieved in each case.

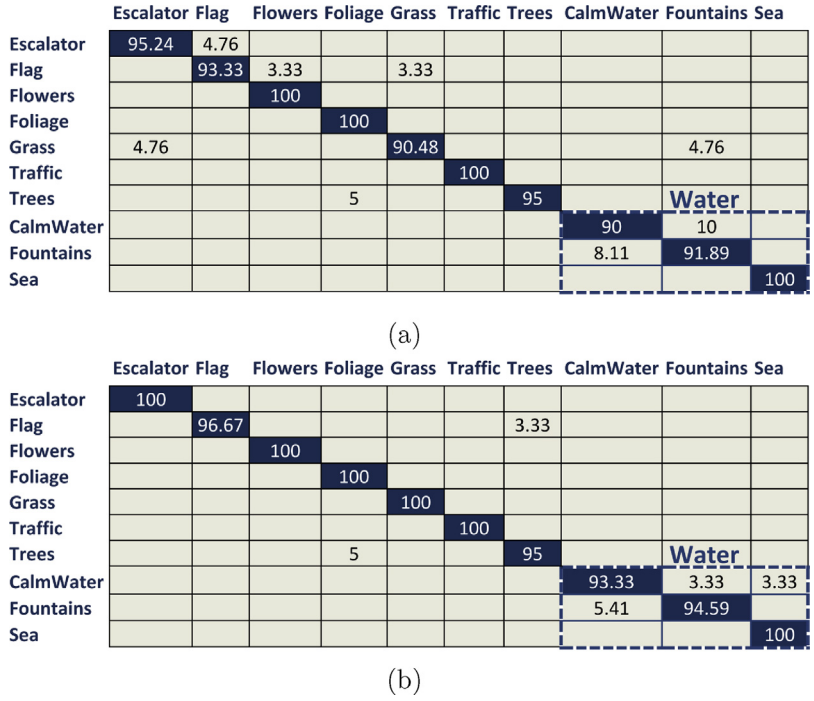

Fig. 7. Multi-class classification accuracy of LBP-flow in gamma split of DynTex Dataset when (a) SVM-based and (b) hybrid method is used.

texture, showing that our proposed framework still detects water-related dynamic scenes robustly. In general, it can be concluded that our algorithm is capable of correctly classifying challenging dynamic textures, distinguishing even between classes with highly correlated dynamics, since it does not perform recognition based only on motion characteristics.

Two additional categories of recognition experiments are carried out for the *Dyntex* dataset: (a) binary classification between water and non-water classes, to compare with corresponding results provided in [16], (b) intra-class classification for water-related dynamic textures, to compare with [7]. Specifically, we compare the binary classification accuracy of our descriptor for the water and non-water classes in Dyntex, with [14,15,36], and [16]. The results for this binary classification are provided in Table 5, where we see that both of our proposed schemes, involving LBP-flow, surpass the first three works [14,15,36], while giving comparable results with that of Mettes et al. [16]. However, it should be noted that in [16] the implementation is highly tailored to water videos, through pre-processing of the data, justifying its high score in binary classification. Intra-class comparisons with the work of [7] in exclusively water related multi-class recognition are also shown in Table 2 and confirm our algorithm's high performance, as it surpasses the recognition accuracy of [7] for most categories of water-related dynamic textures. Even though, during recognition tasks, feature extraction usually constitutes the most time-consuming step, in our case, the extraction of LBP-flow features for the DynTex database, required only about 2.65 fps.

Table 2
Comparisons with SoA for water related classes in Dyntex dataset.

Method	Fountain	Calm water	Sea	Home water	All
HOGP [7]	**88.0%**	81.0%	81.0%	–	–
LBP-flow	55.6%	85%	95.8%	**88.0%**	75.2%
LBP-flow + NN	77.8%	**100%**	**100%**	84.0%	**88.8%**

Bold values indicate the highest score achieved in each case.

5.3.2. DT recognition: the moving vistas dataset

Moving vistas was introduced in [8] and it is the most challenging dataset of all, as it contains video samples of low quality using a moving camera, different viewpoints and significant illumination changes. The multi-class recognition accuracy of LBP-flow was estimated and compared with the SoA on scene recognition in [12] and [8]. The results, depicted in Table 3, show that our hybrid scheme achieves significantly better recognition rates compared to the SoA for the multi-classification task, with detailed classification accuracy for each class provided in Fig. 8.

The binary classification performance of our method was also validated on this dataset, for the water class, with its results presented in Table 3. As seen, our proposed scheme achieves highly accurate outcomes, with a rate of 73.1% and 84.6% for the SVM and NN-based frameworks respectively. However, due to the lack of relevant results in SoA for Moving vistas([12,8]), possibly due to that dataset's limited size, we construct an approximate binary classification metric to evaluate our binary model. More specifically, we isolate the water-related classes (Fountain, Iceberg

Table 3
Multi and binary recognition accuracy on moving vistas dataset.

Score	Derpanis et al. [12]	Shroff et al. [8]	LBP-flow + SVM	LBP-flow + NN
Multi-class	41.0%	52.0%	62.3%	**67.7%**
Binary	–	–	73.1%	**84.6%**
Approximate binary	32.0 %	52.0 %	72.0 %	**72.0%**

Bold values indicate the highest score achieved in each case.

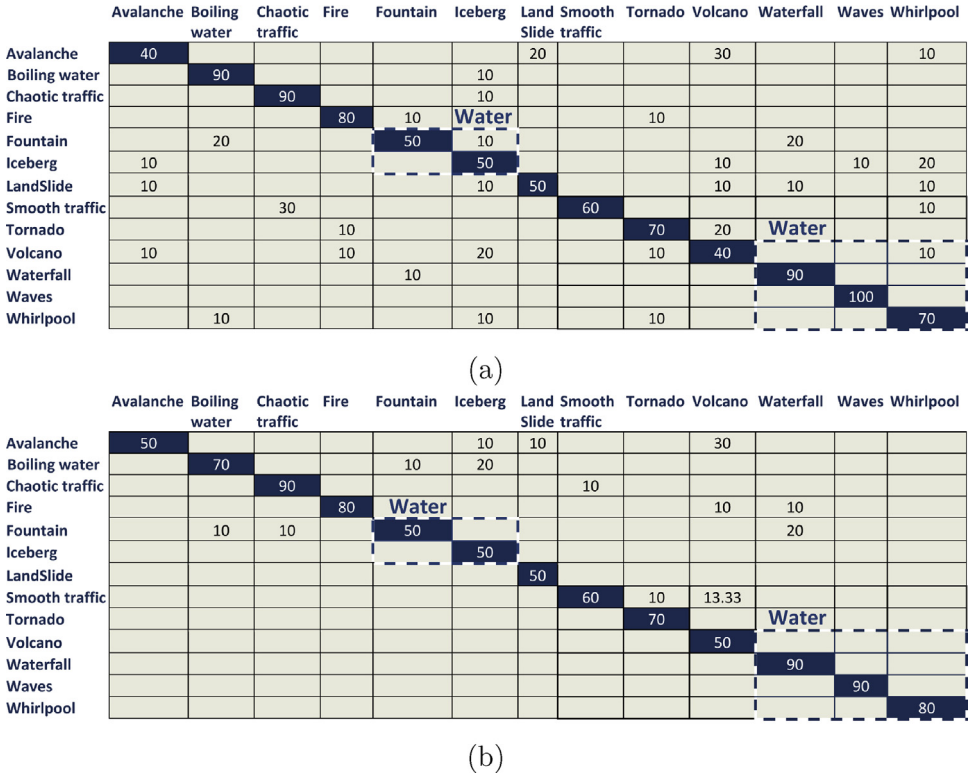

Fig. 8. Multi-class classification accuracy of LBP-flow in movingVistas Dataset with (a) SVM and (b) NN-based classification. In both case, the average classification accuracy for water-related classes is 72%.

Collapse, Waterfall, Waves and Whirlpool) from the confusion matrices provided in [12,8], as well as ours (Fig. 8) and we then estimate their average accuracy, as an "approximate binary accuracy". As shown in the last row of Table 3, the average water-classification accuracy in [12,8] is 32% and 52% respectively, while both our SVM and NN-based schemes both reach 72%, surpassing the SoA. It should be noted that the "average binary accuracy" score achieved by our approach in this case is expected to be lower than that derived from the application of direct binary classification (73.1% for SVM based classification and 84.6% for NN

Table 4
Comparisons with SoA in YUPENN dataset for all classes.

Scene classes	BoST [37]	TSVQ [38]	SOE [12]	Color [39]	GIST [9]	HOF [40]	Chaos [8]	Ours SVM/NN
beach	83	63	87	50	90	37	27	**87.5**/83.3
c. street	90	70	83	47	50	83	17	**100/100**
elevator	100	73	67	83	53	93	40	**100/100**
f. fire	100	80	83	47	50	67	50	**87.5**/83.3
fountain	67	37	47	13	40	30	7	**87.5**/87.0
highway	87	73	77	30	47	33	17	87.5/**95.7**
l. storm	100	80	90	83	57	47	37	**91.7**/66.7
ocean	90	80	100	73	93	60	43	**91.7/91.7**
railway	80	73	87	43	50	83	3	87.5/**95.0**
r. river	80	73	93	57	63	37	3	91.7/**95.8**
sky	93	77	90	30	90	83	33	**95.8/95.8**
snowing	83	77	33	53	20	57	10	87.5/**100**
waterfall	67	53	43	30	33	60	10	91.7/**95.8**
w. farm	77	57	57	57	47	33	17	91.7/**100**
Avg.(%)	85	69	74	50	56	59	20	91.4/**92.2**

Bold values indicate the highest score achieved in each case.

based classification, respectively), as the constructed multi-class model is more complex than the binary one. However, it is safe to infer that since our algorithm outperforms the SoA on water-related classes for the complex multi-classification process, it is likely that its performance will be superior to SoA for the simpler binary classification task as well.

Similarly to the Dyntex data, the cost of feature extraction in the low resolution Moving vistas dataset is kept quite low, requiring about 9.7 fps. This low computational cost makes proposed method appropriate for near real time monitoring in surveillance applications, where videos are often recorded at 7–8 fps.

5.3.3. DT recognition: the YUPENN dataset

YUPENN [12] comprises of 420 videos, mainly of low quality, from 14 different classes. It constitutes a challenging dataset, as each class is represented by a limited number of videos of short duration, ranging from 37 up to 180 frames. Despite these drawbacks, experiments on multi-classification tasks were conducted for all classes, with the results depicted in Table 4. In these experiments, our method is compared with many approaches from the SoA, also reported in [37,12]. More precisely, we compare our results to those of: the bag-of-systems Tree (BoST) representation [37], the tree-structured vector quantization (TSVQ) [38], the spatiotemporal oriented energy (SOE) features [12], the combined use of oriented boundaries and surface color information [39], the Spatial Envelope representation [9], the Histogram of Oriented Flow descriptors [40] and finally the use of the theory of chaotic systems [8] to capture scene dynamics. It is clear from Table 4 that our proposed framework achieves remarkable accuracy rates for all classes, near or above 90% for both shallow and hybrid implementations, outperforming the SoA in all cases.

5.3.4. DT recognition: the VideoWaterDatabase dataset

The VideoWatetDatabase [16] is a very comprehensive database of dynamic textures of the water/non-water classes. It contains a large number of high resolution videos, with a sufficient amount of training data, as well as ground truth for the water regions, in the form of binary masks. Other benchmarking datasets for dynamic textures do not contain as many training videos and ground truth, making this dataset most appropriate for in depth evaluation of our dynamic texture recognition and localization methods.

The *VideoWaterDatabase* introduced in [16] consists of 260 high definition videos, where the presence of water needs to be detected. This dataset contains water and non-water samples from 7 and 5 classes respectively. The patterns between the two classes are quite similar and very difficult to model. Comparisons with other dynamic texture modeling methods based on LBP are provided in Table 5, where the method is compared against 4 other SoA works, which use different approaches for texture representation. These include: Volume Local Binary Patterns (VLBP) used by [14], LBP-TOP descriptors of [15], the transferred ConvNet Feature (TCoF) of [36] and the LBP-Fourier descriptor applied in [16]. As shown in Table 5, our approach leads to the highest accuracy, of 98.8%, providing robust recognition results in this case as well. For the VideoWaterDatabase videos, the extraction of LBP-Flow features required about 2.05 fps, showing that our method is indeed computationally efficient.

Table 5
Comparisons with SoA for water recognition.

Method	VideoWaterDatabase	Dyntex
LBP-Fourier [16]	98.4%	**95.8%**
VLBP [14]	93.8%	90.0%
LBP-TOP [15]	93.3%	87.5%
st-TCoF [36]	97.2%	90.0%
LBP-flow	98.3%	92.74%
LBP-flow +NN	**98.8%**	94.35%

Bold values indicate the highest score achieved in each case.

5.4. Dynamic texture localization

The localization of dynamic textures is very challenging, due to their highly non-rigid nature, the transparency that often characterizes them, numerous occlusions and the complexity of the motions in them. We have carried out experiments aiming to localize dynamic textures, related to outdoors crisis scenarios, but without tailoring our approach to a specific texture. Thus, in order to reinforce our algorithm's general applicability, we avoided applying any pre-processing and post-processing steps.

5.4.1. Water localization

We initially carried out the localization of water-related dynamic textures on the challenging VideoWaterDatabase (VWD) and DynTex datasets, where 32×32 and 16×16 grids of superpixels have been deployed for the bottom and top layer respectively. Results for VWD are provided in Table 6, where it can be seen that our method reaches the remarkable accuracy of 96.7% and 96.15% for the SVM and NN frameworks respectively, significantly surpassing that of [16].

We also evaluate our algorithm's performance on the DynTex dataset by first splitting the videos into water and non-water categories, with 2/3 of them used for training. Our localization score in this case was 86.1% comparable to that provided in [16], despite DynTex's limited training size for water class. It should be emphasized that, in contrast to our general framework, the algorithm of [16] is designed for detecting and localizing exclusively water-based dynamic textures. Specifically, they carry out several preprocessing steps concerning water properties, such as the removal of water reflections from water ripples, so as to obtain water regions of a homogeneous appearance. They also add a post-processing step, to further improve their accuracy. As a result, they achieve accurate results for water detection, but their method cannot be extended to the general case of texture recognition.

We further investigate our proposed localization framework and compare its results in the cases where: (1) a smaller window of $W_{LBP} = 10$ frames is used, and (2) Histogram of oriented gradients-Histogram of optical flow (HOG/HOF) descriptor is applied instead of the LBP-flow for a time window of $W_{HOG/HOF} = 10$ and $W_{HOG/HOF} = 30$ frames respectively. Our results are depicted in Table 7. As observed, LBP-flow descriptor outmatches the classic HOG/HOF descriptor in the texture localization task, proving LBP-flow's suitability for dynamic texture representation. Furthermore, even a small window of $W_{LBP} = 10$ seems to be adequate for our algorithm's efficient performance, however a larger window of 30 frames is finally preferred, as it leads to more accurate results.

Instances of the localization process for VWD are provided in Fig. 9. As it can be seen, our multi-resolution scheme succeeds in capturing local non-water areas occupying only a small part of the frame (a), (e), while at the same time challenging water scenes containing shadows and running water (c), (h) are also correctly

Table 6
Water localization in VWD and DynTex.

	LBP-Fourier [16]	LBP-flow (SVM)	LBP-flow (NN)
VWD	92.3%	**96.7%**	96.15%
DynTex	**87.9%**	83.4%	86.1%

Bold values indicate the highest score achieved in each case.

Table 7
Water localization in VWD for different temporal window sizes.

Method	Score
LBP-flowW30	**96.7%**
LBP-flowW10	94.27%
HOG/HOFW10	91%
HOG/HOFW30	92.22%

Bold values indicate the highest score achieved in each case.

localized. The minor errors of our algorithm can be attributed to its general non-water based nature, and the omission of any post-processing steps which would smooth the final results. Sample videos, as derived from the direct application of the machine learning model of the proposed localization framework, without any pre-processing or post processing can be found in the following link: *https://vimeo.com/channels/1323363/*. Transparent blue and red color are used to depict the areas where each dynamic texture of interest was detected by our algorithm. As seen, water video sequences containing shadows (e.g. lake) or different motion dynamics (e.g. fountain) are correctly localized as cohesive water areas, in contrary to the segmentation that we would expect from the SoA segmentation-based works, as discussed in Section 2.

The computational cost of the localization task is quite low, at 0.23 frames/sec, despite the fact that the algorithm is superpixel oriented, meaning that it infers about each superpixel's texture separately. It should also be noted that our algorithm is executed online, with only the past 30 frames being taken into account, making it appropriate for real world outdoor applications.

5.4.2. Fire localization

We also examine our method's efficacy in localization task for the fire texture, using videos from the YUPENN dataset. The fire, a texture of high interest for outdoors surveillance, constitutes a challenging dynamic texture as the boundaries between fire and smoke are often difficult to discern, even for human observers, while the wind constantly changes the flames' shape and direction. Despite this fact, our method extracts a mask at every 30 frames, which captures most of the fire and flames. In our experiments, the dataset is split into fire and non-fire scenes, with the first category containing exclusively the class Forest fire, as it is the only class related to fire hazard, while the other 13 classes are classified as non-fire. The large deviation between the number of videos representing each category, with 30 videos of the fire class compared to 390 videos of the non-fire class, hinder the proper evaluation of our method in this task. Nonetheless, we still

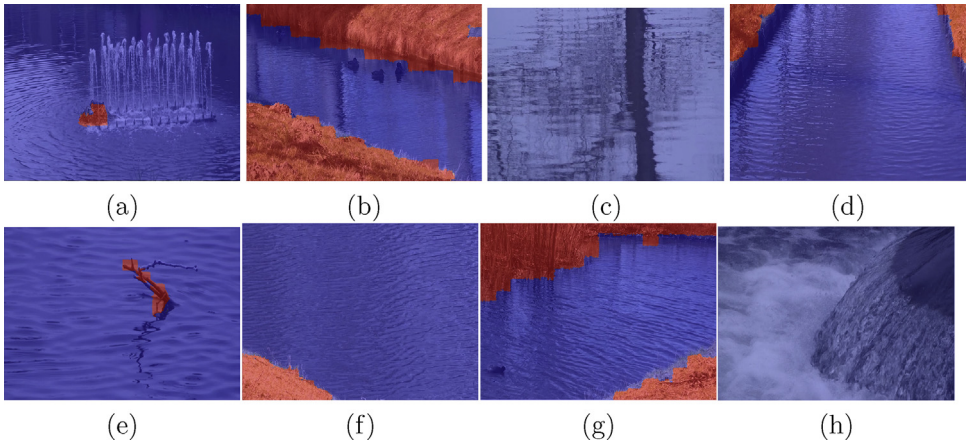

Fig. 9. Instances of water localization in VideoWaterDatabase dataset. Blue color depicts the region that water was detected from our algorithm, while red color stands for the non-water regions. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of the article.)

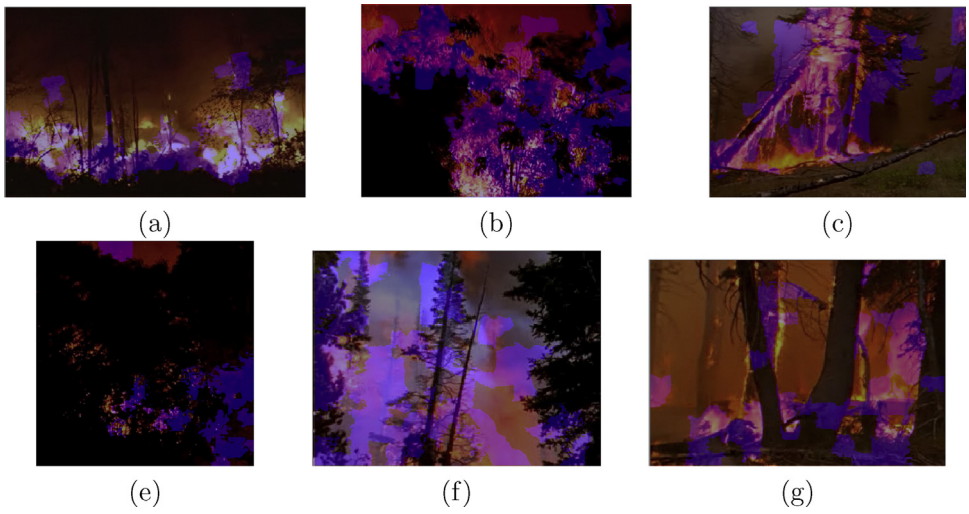

Fig. 10. Instances of fire localization in Yupenn dataset. Blue color depicts the region that fire was detected from our algorithm. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of the article.)

examine these videos to assess our algorithm's generality, using 3/4 of the fire videos for training, and random videos of all the other classes for the non-fire category. We choose these videos in a 3 non-fire-to-1 fire ratio, so as to complete the training set.

Qualitative results of our algorithm's performance on the test videos are presented in Fig. 10, where transparent blue color is used to depict regions where fire was detected by our method. The variations in fire texture and appearance are captured in most cases, demonstrating that our method can be effectively applied on different textures. Some false alarms are observed, however these can be attributed to the lack of training data. There is still a lack of complete benchmarking datasets for dynamic textures, with the number of the available fire videos in our case being inadequate and of short duration. Specifically, the training videos of fire comprise of only 37 or 120 frames, which in many cases significantly reduce the ability of our method to extract sufficient descriptors. Video samples of fire localization may be found in *https://vimeo.com/channels/1323363/*.

Results in the challenging fire class, prove that our algorithm can efficiently cope with other types of dynamic textures, as expected, since its design is not tailored to a specific category of dynamic texture videos. Our framework is thus able to provide a powerful model capable of discriminating between different classes, despite their various dynamics and inter-class similarities.

6. Conclusion

In this work we propose a novel framework acting on both the local and global scale for dynamic texture recognition and localization. Informative regions, retaining details of the dynamic textures, while avoiding overfitting to local noise, are extracted by clustering superpixels, which accurately detail the boundaries of dynamic textures. An LBP-flow descriptor is then combined with Fisher encoding and a Neural Network, forming a novel, powerful framework, capable of capturing and describing a wide range of dynamic scenes. Its remarkable performance on several different benchmark datasets, for binary and multiple classification, proves our method's generality. Its excellent performance on water localization, makes the proposed scheme suitable for real life situations, where image details also count for the proper handling of a potential hazard. Given its general applicability, by design, the proposed method can be deployed for the detection and localization of various types of dynamic textures that occur in outdoor environments, while its low computational cost makes it appropriate for a variety of real world applications.

Acknowledgements

This work was funded by the European Unions Horizon 2020 Research and Innovation Programme, within the project “beAWARE: Enhancing decision support and management services in extreme weather climate events”, under grant agreement No. 7000475.

References

[1] K. Avgerinakis, P. Giannakeris, A. Briassouli, A. Karakostas, S. Vrochidis, I. Kompatsiaris, Lbp-flow and hybrid encoding for real-time water and fire classification, IEEE International Conference on Computer Vision Workshop (ICCVW) (2017).
[2] M. Fritz, B. Leibe, B. Caputo, B. Schiele, Integrating representative and discriminant models for object category detection, Tenth IEEE International Conference on Computer Vision, ICCV, vol. 2, IEEE, 2005, pp. 1363–1370.
[3] G. Doretto, A. Chiuso, Y.N. Wu, S. Soatto, Dynamic textures, Int. J. Comput. Vis. 51 (2) (2003) 91–109.
[4] A.B. Chan, N. Vasconcelos, Modeling, clustering, and segmenting video with mixtures of dynamic textures, IEEE Trans. Pattern Anal. Mach. Intell. 30 (5) (2008) 909–926.
[5] A.B. Chan, N. Vasconcelos, Layered dynamic textures, IEEE Trans. Pattern Anal. Mach. Intell. 31 (10) (2009) 1862–1879.
[6] A. Mumtaz, E. Coviello, G.R.G. Lanckriet, A.B. Chan, Clustering dynamic textures with the hierarchical em algorithm for modeling video, IEEE Trans. Pattern Anal. Mach. Intell. 35 (7) (2013) 1606–1621.
[7] K. Dimitropoulos, P. Barmpoutis, A. Kitsikidis, N. Grammalidis, Classification of multidimensional time-evolving data using histograms of Grassmannian points, IEEE Trans. Circuits Syst. Video Technol. PP (99) (2017) 1.
[8] N. Shroff, P. Turaga, R. Chellappa, Moving vistas: exploiting motion for describing scenes, IEEE Computer Society Conference on Computer Vision and Pattern Recognition (2010) 1911–1918.
[9] A. Oliva, A. Torralba, Modeling the shape of the scene: a holistic representation of the spatial envelope, Int. J. Comput. Vis. 42 (3) (2001) 145–175.
[10] J. Chen, G. Zhao, M. Salo, E. Rahtu, M. Pietikainen, Automatic dynamic texture segmentation using local descriptors and optical flow, IEEE Trans. Image Process. 22 (1) (2013) 326–339.
[11] V. Kaltsa, A. Briassouli, I. Kompatsiaris, L.J. Hadjileontiadis, M.G. Strintzis, Swarm intelligence for detecting interesting events in crowded environments, IEEE Trans. Image Process. 24 (7) (2015) 2153–2166.
[12] K.G. Derpanis, M. Lecce, K. Daniilidis, R.P. Wildes, Dynamic scene understanding: the role of orientation features in space and time in scene classification, IEEE Conference on Computer Vision and Pattern Recognition (2012) 1306–1313.
[13] C. Feichtenhofer, A. Pinz, R.P. Wildes, Bags of spacetime energies for dynamic scene recognition, The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2014).
[14] G. Zhao, M. Pietikainen, Local binary pattern descriptors for dynamic texture recognition, 18th International Conference on Pattern Recognition (ICPR'06), vol. 2 (2006) 211–214.
[15] G. Zhao, T. Ahonen, J. Matas, M. Pietikainen, Rotation-invariant image and video description with local binary pattern features, IEEE Trans. Image Process. 21 (4) (2012) 1465–1477.
[16] P. Mettes, R.T. Tan, R.C. Veltkamp, Water detection through spatio-temporal invariant descriptors, Comput. Vis. Image Understand. 154 (2017) 182–191.
[17] T. Amiaz, S. Fazekas, D. Chetverikov, N. Kiryati, Detecting Regions of Dynamic Texture, Springer Berlin Heidelberg, Berlin, Heidelberg, 2007, pp. 848–859.
[18] L. Chen, Y. Qiao, Markov random field based dynamic texture segmentation using inter-scale context, IEEE International Conference on Information and Automation (ICIA) (2016) 1924–1927.
[19] L. Wang, D.-C. He, Texture classification using texture spectrum, Pattern Recogn. 23 (8) (1990) 905–910.
[20] L. Liu, L. Zhao, Y. Long, G. Kuang, P. Fieguth, Extended local binary patterns for texture classification, Image Vis. Comput. 30 (2) (2012) 86–99.
[21] X. Qian, X.-S. Hua, P. Chen, L. Ke, Plbp: an effective local binary patterns texture descriptor with pyramid representation, Pattern Recogn. 44 (10) (2011) 2502–2515 semi-Supervised Learning for Visual Content Analysis and Understanding.
[22] C. Shan, S. Gong, P.W. McOwan, Facial expression recognition based on local binary patterns: a comprehensive study, Image Vis. Comput. 27 (6) (2009) 803–816.
[23] T. Ahonen, A. Hadid, M. Pietikainen, Face description with local binary patterns: application to face recognition, IEEE Trans. Pattern Anal. Mach. Intell. 28 (12) (2006) 2037–2041.
[24] F. Perronnin, J. Sánchez, T. Mensink, Improving the Fisher Kernel for Large-Scale Image Classification, Springer Berlin Heidelberg, Berlin, Heidelberg, 2010, pp. 143–156.
[25] K. Avgerinakis, A. Briassouli, Y. Kompatsiaris, Activity detection using sequential statistical boundary detection (ssbd), Comput. Vis. Image Understand. 144 (2016) 46–61 Individual and Group Activities in Video Event Analysis.
[26] C.R. de Souza, A. Gaidon, E. Vig, A.M. López, Sympathy for the Details: Dense Trajectories and Hybrid Classification Architectures for Action Recognition, Springer International Publishing, Cham, 2016, pp. 697–716.
[27] R. Achanta, A. Shaji, K. Smith, A. Lucchi, P. Fua, S. Süsstrunk, Slic superpixels compared to state-of-the-art superpixel methods, IEEE Trans. Pattern Anal. Mach. Intell. 34 (11) (2012) 2274–2282.
[28] R. Péteri, S. Fazekas, M.J. Huiskes, Dyntex: a comprehensive database of dynamic textures, Pattern Recogn. Lett. 31 (12) (2010) 1627–1632 pattern Recognition of Non-Speech Audio.
[29] P. Mettes, R.T. Tan, R. Veltkamp, On the segmentation and classification of water in videos, International Conference on Computer Vision Theory and Applications (VISAPP), vol. 1 (2014) 283–292.
[30] G. Farnebäck, Two-Frame Motion Estimation Based on Polynomial Expansion, Springer Berlin Heidelberg, Berlin, Heidelberg, 2003, pp. 363–370.
[31] S. Dubois, R. Péteri, M. Ménard, Characterization and recognition of dynamic textures based on the 2d+t curvelet transform, Signal Image Video Process. 9 (4) (2015) 819–830.
[32] J.R. Smith, C.-Y. Lin, M. Naphade, Video texture indexing using spatio-temporal wavelets, Proceedings. International Conference on Image Processing, vol. 2 (2002) II-437–II-440.
[33] Y. Xu, Y. Quan, Z. Zhang, H. Ling, H. Ji, Classifying dynamic textures via spatiotemporal fractal analysis, Pattern Recogn. 48 (10) (2015) 3239–3248 Discriminative Feature Learning from Big Data for Visual Recognition.
[34] Y. Xu, S. Huang, H. Ji, C. Fermüller, Scale-space texture description on sift-like textons, Comput. Vis. Image Understand. 116 (9) (2012) 999–1013.

[35] H. Ji, X. Yang, H. Ling, Y. Xu, Wavelet domain multifractal analysis for static and dynamic texture classification, IEEE Trans. Image Process. 22 (1) (2013) 286–299.
[36] X. Qi, C.-G. Li, G. Zhao, X. Hong, M. Pietikäinen, Dynamic texture and scene classification by transferring deep image features, Neurocomputing 171 (2016) 1230–1241.
[37] A. Mumtaz, E. Coviello, G.R.G. Lanckriet, A.B. Chan, A scalable and accurate descriptor for dynamic textures using bag of system trees, IEEE Trans. Pattern Anal. Mach. Intell. 37 (4) (2015) 697–712.
[38] D. Nister, H. Stewenius, Scalable recognition with a vocabulary tree, IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06), vol. 2 (2006) 2161–2168.
[39] S. Grossberg, T.-R. Huang, Artscene: a neural system for natural scene classification, J. Vis. 9 (4) (2009) 6.
[40] M. Marszalek, I. Laptev, C. Schmid, Actions in context, IEEE Conference on Computer Vision and Pattern Recognition (2009) 2929–2936.